数据结构

本文档主要介绍智能制造业企业评级系统开发中所用到的数据结构。

0 文档标识

作者	日期	版本
袁易锋	2021-03-01	v1.0

1 Enterprise模块

1.1 Enterprise结构

```
public class Enterprise implements Serializable {

    /**
     * 数据库主键
     */
    private Integer id;

    /**
     * 用户类型，0代表普通用户，1代表企业用户
     */
    private Integer userType;

    /**
     * 企业的评级，默认0为最高
     */
    private Integer eGrade;

    /**
     *   企业名字
     */
    private String name;

    /**
     * 注册号
     */
    private String registerNumber;

    /**
     * 联系人名字
     */
    private String contactName;

    /**
     * 邮箱
     */
    private String email;

    /**
     * 营业执照
     */
    private String license;

    /**
     * 联系号码
     */
    private String contactNumber;

    /**
     * 是否有效，情况是否属实
     */
    private boolean valid;

    /**
     * 企业密码
     */
    private String password;

    /**
     * 企业所处行业类型，共有5种，分别是离散大批量行业
     */
    private Integer type;

    /**
     * 企业照片
     */
    private String ePhoto;

    /**
     * 财务指标评分
     */
    private Double financialScore;

    /**
     * 两化指标评分
     */
    private Double diverScore;

    /**
     * 总信用得分
     */
    private  Double totalScore;

}
```

1.2 financial指标

包括“FinancialIndex”初始指标和“FinancialPercent”当前指标

```
public class FinancialIndex {

    public FinancialIndex(Integer id, Double[] args){
        if (args.length != 18){
            System.out.println("Initialization failure");
            return;
        }
        this.enterprise_id = id;
        this.ttm = args[0];
        this.debt_ratio = args[1];
        this.current_ratio = args[2];
        this.quick_ratio = args[3];
        this.cf_cl = args[4];
        this.ttal = args[5];
        this.operating_revenue = args[6];
        this.total_profit = args[7];
        this.total_liabilities = args[8];
        this.roe = args[9];
        this.profit_rate = args[10];
        this.gross_profit = args[11];
        this.fe_rev = args[12];
        this.turnover_days = args[13];
        this.current_assets = args[14];
        this.accounts_receivable = args[15];
        this.ebit_rev = args[16];
        this.industry_prosperity = args[17];
    }

    private Integer enterprise_id;

    private double ttm;

    private double debt_ratio;

    private double current_ratio;

    private double quick_ratio;

    private double cf_cl;

    private double ttal;

    private double operating_revenue;

    private double total_profit;

    private double total_liabilities;

    private double roe;

    private double profit_rate;

    private double gross_profit;

    private double fe_rev;

    private double turnover_days;

    private double current_assets;

    private double accounts_receivable;

    private double ebit_rev;

    private double industry_prosperity;
```

```
}
```

1.3 integrate

1.3.1 MassDiscrete离散大批量企业两化指标

```
public class MassDiscrete {

    public MassDiscrete (Integer id, Double[] args){
        if (args.length != 17) {
            System.out.println("Initialization failure");
            return;
        }
        this.enterprise_id = id;
        this.on_time_delivery_rate = args[0];
        this.system_manage_level = args[1];
        this.emergency_response = args[2];
        this.budget_management = args[3];
        this.auto_instruction_content = args[4];
        this.environmental_management = args[5];
        this.energy_management = args[6];
        this.social_contribution = args[7];
        this.business_collaboration = args[8];
        this.value_network_collaboration = args[9];
        this.tracking_and_feedback = args[10];
        this.maintenance_investment = args[11];
        this.organizational_model = args[12];
        this.plan_implementation = args[13];
        this.decision_support = args[14];
        this.equipment_management = args[15];
        this.hr_management = args[16];
    }

    public MassDiscrete(){}

    private Integer enterprise_id;

    private double on_time_delivery_rate;

    private double system_manage_level;

    private double emergency_response;

    private double budget_management;

    private double auto_instruction_content;

    private double environmental_management;

    private double energy_management;

    private double social_contribution;
    private double business_collaboration;

    private double value_network_collaboration;

    private double tracking_and_feedback;

    private double maintenance_investment;

    private double organizational_model;

    private double plan_implementation;

    private double decision_support;

    private double equipment_management;

    private double hr_management;


}
```

1.3.2 MixIndustry混合行业企业两化指标

```
public class MixIndustry {


    public MixIndustry (Integer id, Double[] args){
        if (args.length != 21){
            System.out.println("Initialization failure");
            return;
        }
        this.enterprise_id = id;
        this.system_manage_level = args[0];
        this.test_data_rate = args[1];
        this.tracking_oforders = args[2];
        this.online_purchasing_rate = args[3];
        this.monitoring_coverage = args[4];
        this.auto_instruction_content = args[5];
        this.collection_of_energy_data = args[6];
        this.environmental_management = args[7];
        this.info_exchange_and_sharing = args[8];
        this.value_network_collaboration = args[9];
        this.green_development = args[10];
        this.social_contribution = args[11];
        this.information_investment = args[12];
        this.equipment_assets = args[13];
        this.organizational_model = args[14];
        this.informatization_plan_implementation = args[15];
        this.material_requirements = args[16];
        this.business_scope = args[17];
        this.equipment_management = args[18];
        this.hr_management = args[19];
        this.industrial_software_innovation = args[20];
    }
    private Integer enterprise_id;

    private double system_manage_level;

    private double test_data_rate;

    private double tracking_oforders;

    private double online_purchasing_rate;

    private double monitoring_coverage;

    private double auto_instruction_content;

    private double collection_of_energy_data;

    private double environmental_management;

    private double info_exchange_and_sharing;

    private double value_network_collaboration;

    private double green_development;

    private double social_contribution;

    private double information_investment;

    private double equipment_assets;

    private double organizational_model;

    private double informatization_plan_implementation;

    private double material_requirements;

    private double business_scope;

    private double equipment_management;

    private double hr_management;

    private double industrial_software_innovation;

}
```

1.3.3 ProcIndustry过程化行业两化指标

```
public class ProcIndustry {


    public ProcIndustry (Integer id, Double[] args){
        if (args.length != 25){
            System.out.println("Initialization failure");
            return ;
        }
        this.enterprise_id = id;
        this.system_manage_level = args[0];
        this.budget_management = args[1];
        this.auto_upload_info_range = args[2];
        this.auto_instruction_content = args[3];
        this.auto_information_coverage = args[4];
        this.auto_instruction_situation = args[5];
        this.integrated_management_cover = args[6];
        this.online_purchasing_rate = args[7];
        this.min_scheduling_unit = args[8];
        this.energy_management = args[9];
        this.social_contribution = args[10];
        this.business_collaboration = args[11];
        this.info_exchange_and_sharing = args[12];
        this.value_network_collaboration = args[13];
        this.green_development = args[14];
        this.maintenance_investment = args[15];
        this.organizational_model = args[16];
        this.plan_implementation = args[17];
        this.material_requirements = args[18];
        this.research_development = args[19];
        this.business_scope = args[20];
        this.equipment_management = args[21];
        this.hr_management = args[22];
        this.industrial_software_innovation = args[23];
        this.output_value = args[24];
    }

    private Integer enterprise_id;

    private double system_manage_level;

    private double budget_management;

    private double auto_upload_info_range;

    private double auto_instruction_content;

    private double auto_information_coverage;

    private double auto_instruction_situation;

    private double integrated_management_cover;

    private double online_purchasing_rate;

    private double min_scheduling_unit;

    private double energy_management;

    private double social_contribution;

    private double business_collaboration;

    private double info_exchange_and_sharing;

    private double value_network_collaboration;
    private double green_development;

    private double maintenance_investment;

    private double organizational_model;

    private double plan_implementation;

    private double material_requirements;

    private double research_development;

    private double business_scope;

    private double equipment_management;

    private double hr_management;

    private double industrial_software_innovation;

    private double output_value;

}
```

1.3.4 ServeIndustry服务行业两化指标

```
public class ServeIndustry {


    public ServeIndustry (Integer id, Double[] args){
        if (args.length != 18){
            System.out.println("Initialization failure");
            return;
        }
        this.enterprise_id = id;
        this.system_manage_level = args[0];
        this.call_center = args[1];
        this.budget_management = args[2];
        this.service_Integration = args[3];
        this.online_purchase_rate = args[4];
        this.application_scope_of_purchasing_ecommerce = args[5];
        this.labor_productivity = args[6];
        this.Informatization_energy_management = args[7];
        this.Industry_chain_cooperation = args[8];
        this.Information_exchange_and_sharing = args[9];
        this.maintenance_investment_proportion = args[10];
        this.enterprise_level_unified_coding = args[11];
        this.enterprise_organization_mode = args[12];
        this.informatization_plan_implementation = args[13];
        this.informatization_decision_support = args[14];
        this.informatization_equipment_management = args[15];
        this.informatization_human_resource_management = args[16];
        this.cloud_platform_usage = args[17];
    }

    private Integer enterprise_id;

    private double system_manage_level;

    private double call_center;

    private double budget_management;

    private double service_Integration;

    private double online_purchase_rate;

    private double application_scope_of_purchasing_ecommerce;

    private double labor_productivity;

    private double Informatization_energy_management;

    private double Industry_chain_cooperation;

    private double Information_exchange_and_sharing;

    private double maintenance_investment_proportion;

    private double enterprise_level_unified_coding;

    private double enterprise_organization_mode;

    private double informatization_plan_implementation;

    private double informatization_decision_support;

    private double informatization_equipment_management;

    private double informatization_human_resource_management;

    private double cloud_platform_usage;

}
```

1.3.5 SmeDiscrete中小型离散化行业两化指标

```
public class SmeDiscrete {

    public SmeDiscrete (Integer id, Double[] args){
        if (args.length != 26){
            System.out.println("Initialization failure");
            return ;
        }
        this.enterprise_id = id;
        this.system_manage_level = args[0];
        this.financial_system_monitor_sales = args[1];
        this.auto_instruction_content = args[2];
        this.upload_info_specification = args[3];
        this.info_qua_coverage_area = args[4];
        this.info_qua_function = args[5];
        this.info_monitor_outsourceing = args[6];
        this.info_manage_energy = args[7];
        this.social_contribution_rate = args[8];
        this.info_cooperate_business = args[9];
        this.inter_realize_value_network_synergy = args[10];
        this.online_product_ext_recontrol = args[11];
        this.info_inputmoney_five = args[12];
        this.infosys_rate_info = args[13];
        this.total_assets_proequ = args[14];
        this.enterprise_organization_model = args[15];
        this.info_plan_implement = args[16];
        this.info_matdemand_plan_management = args[17];
        this.info_app_design = args[18];
        this.info_analyze_business_scope = args[19];
        this.info_equ_manage_fun_level = args[20];
        this.info_humanres_fun_level = args[21];
        this.industrysoftware_innovation_ability = args[22];
        this.newproduct_develope_cycle = args[23];
        this.patent_ownership_hundred = args[24];
        this.productmodel_define_data = args[25];
    }

    private Integer enterprise_id;

    private double system_manage_level;

    private double financial_system_monitor_sales;

    private double auto_instruction_content;

    private double upload_info_specification;

    private double info_qua_coverage_area;

    private double info_qua_function;

    private double info_monitor_outsourceing;

    private double info_manage_energy;

    private double social_contribution_rate;

    private double info_cooperate_business;

    private double inter_realize_value_network_synergy;

    private double online_product_ext_recontrol;

    private double info_inputmoney_five;

    private double infosys_rate_info;

    private double total_assets_proequ;
    private double enterprise_organization_model;

    private double info_plan_implement;

    private double info_matdemand_plan_management;

    private double info_app_design;

    private double info_analyze_business_scope;

    private double info_equ_manage_fun_level;

    private double info_humanres_fun_level;

    private double industrysoftware_innovation_ability;

    private double newproduct_develope_cycle;

    private double patent_ownership_hundred;

    private double productmodel_define_data;


}
```

1.4 VO

1.4.1 EnterpriseVO

```
public class EnterpriseVO implements Serializable {

    /**
     * 数据库主键
     */
    private Integer id;


    /**
     * 用户类型，0代表普通用户，1代表企业用户
     */
    private Integer userType;

    /**
     * 企业密码
     */
    private String password;

    /**
     *  企业名字
     */
    private String name;

    /**
     * 注册号
     */
    private String registerNumber;

```

```java
    /**
     * 联系人名字
     */
    private String contactName;

    /**
     * 邮箱
     */
    private String email;

    /**
     * 营业执照
     */
    private String license;

    /**
     * 联系号码
     */
    private String contactNumber;

    /**
     * 是否有效
     */
    private boolean valid;

    /**
     * 企业的评级，默认0为最高,-1代表未评分
     */
    private Integer eGrade;

    /**
     * 企业照片
     */
    private String ePhoto;


    /**
     * 企业所处行业类型，共有5种，分别是离散大批量行业
     */
    private Integer type;

    /**
     * 财务指标评分
     */
    private Double financialScore;

    /**
     * 两化指标评分
     */
    private Double diverScore;

    /**
     * 总信用得分
     */
    private  Double totalScore;


}
```

1.4.2 EnterpriseForm

```java
public class EnterpriseForm {

    /**
     * 企业用户登录的邮箱
     */
    private String email;

    /**
     * 企业登录的密码
     */
    private String password;
}
```

1.4.3 EnterpriseTarget

```java
public class EnterpriseTarget {

    public EnterpriseTarget(Double[] div, Double[] fin){
        this.div = div;
        this.fin = fin;
    }

    /**
     * 两化数组
     * */
    private Double[] div;

    /**
     * 财务数组
     * */
    private Double[] fin;

    public Double[] getDiv(){
        return div;
    }
    public Double[] getFin(){
        return fin;
    }
}
```

1.4.4 TargetObject

```
public class TargetObject {

    public TargetObject(Double[] array){
        this.array = array;
    }

    Double[] array;

    public Double[] getArray(){
        return array;
    }
}
```

2 User模块

2.1 User结构

```
public class User {

    private Integer id;

    private String email;

    private String password;

    private String username;

    private String phoneNumber;

    /**
     * 用户类型,0代表用户，1代表企业
     */
    private Integer userType;

    /**
     * 用户头像 Url
     */
    private String avatarUrl;

}
```

2.2 Browse浏览记录

```
public class Browse {

    /**
     * 用户Id
     */
    private Integer userId;

    /**
     * 浏览的企业Id
     */
    private Integer epId;

    /**
     * 浏览的时间
     */
    private String createTime;
}
```

2.3 VO

2.3.1 UserVO

```
public class UserVO implements Serializable {

    private Integer id;

    private String email;

    private String password;

    private String username;

    private String phoneNumber;

    /**
     * 用户类型,0代表用户，1代表企业
     */
    private Integer userType;

    /**
     * 用户头像 Url
     */
    private String avatarUrl;

}
```

2.3.2 BrowseVO

```
public class BrowseVO {
    /**
     * 用户Id
     */
    private Integer userId;

    /**
     * 浏览的企业信息,用于前端展示
     */
    private EnterpriseVO enterpriseVO;

    /**
     * 浏览的时间
     */
    private String createTime;
}
```

2.3.3 UserForm

```
public class UserForm implements Serializable {

    /**
     * 用户邮箱，不可重复
     */
    private String email;
    /**
     * 用户密码
     */
    private String password;

    /**
     * 用户类型，后续 可用枚举类代替
     */
    private String userType;

}
```

3 ResponseVO

供前端解析的结构

```
public class ResponseVO implements Serializable{

    /**
     * 调用是否成功
     */
    private Boolean success;

    /**
     * 返回的提示信息
     */
    private String message;

    /**
     * 返回所携带的对象信息
     */
    private Object content;

    public static ResponseVO buildSuccess(){
        ResponseVO response=new ResponseVO();
        response.setSuccess(true);
        return response;
    }


    public static ResponseVO buildSuccess(Object content){
        ResponseVO response=new ResponseVO();
        response.setContent(content);
        response.setSuccess(true);
        return response;
    }

    public static ResponseVO buildFailure(String message){
        ResponseVO response=new ResponseVO();
        response.setSuccess(false);
        response.setMessage(message);
        return response;
    }

    public static ResponseVO buildFailure(Object content){
        ResponseVO response=new ResponseVO();
        response.setContent(content);
        response.setSuccess(false);
        return response;
    }
}
```

4 补充说明

详细代码请参考项目文件